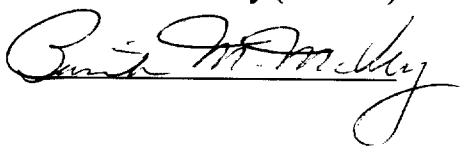
Action Documentation Manager Plus

An Honors Thesis (HONORS 499)
By
Daniel Maddox, Jr.

Dr. Bonita McVey (Advisor)



Ball State University
Muncie, Indiana

May 1998

Graduation: May 1998

Table Of Contents

SECTION I -- ADMP

Introduction

One of the most time consuming tasks of software development is documentation. In an effort to maintain current documentation for large software projects, automating the task of creating documentation is essential. A large software project may require years of development time followed by years of system maintenance and upgrades. To be successful, such a project requires close communication between team members and thorough technical documentation. The necessity of such documentation becomes clear in the second or third generation of the development team. At this point, most of the original developers have moved on to other projects or to other companies and are not available to help recapture the original design of the project. Without good technical documentation, much of the original thinking that went into the project may be lost.

Action Documentation Manager Plus (ADMP) version 1.0 is a tool designed to help produce and manage technical documentation for projects written in C with embedded SQL. The goal of ADMP is to capture project documentation based on the source code and on the comments found within the source code. ADMP captures this documentation and stores it in a database which can be used by software developers to lookup information on one or more projects and to print reports which help them in overseeing the development process.

For each project, ADMP tracks the following:

- The location of the project.
- All files included in the project.
- All functions included in the project.
- All tables used by the project.
- The total count of non-comment/non-blank lines in the project.
- The total count of comment lines in the project.

For each function, ADMP tracks the following information:

- The name of the function.
- A comment block explaining the purpose of the function.
- The file in which the function can be found.
- The line on which the function can be found.
- All projects which include the function.
- All tables the function uses in select , delete, insert, or update statements.
- The count of non-comment/non-blank lines for the function.
- The count of comment lines for the function.

Experience with small and medium software development projects has taught me both the rewards and the challenges of software development. ADMP has been a capstone project for me because it has helped me to discover new ways to deal with the challenges of software development. In addition, I have designed ADMP with flexibility in mind. Enhancing ADMP to generate more project data should be relatively easy.

The Future of ADMP

It is important to note that ADMP is a great starting point. Based on the data ADMP generates, reports to summarize the following could easily be created:

- Any functions included in a project that are included in other projects.
- Any functions that are defined in more than one place.
- Any tables that are never used.
- All functions that do not use any tables.
- All functions that use at least one table.
- All functions without comments.
- All functions longer than X lines.
- All functions without header comment blocks.

Future plans for ADMP include modifying it to track each of the following:

- All references to a function.
- All columns a function uses in select, update, delete, or insert statements.
- All modifications made to a function.

Using ADMP

ADMP has two parts. The first part is a command line program which is used to refresh project documentation. This command line program can be scheduled to run periodically to refresh data for any or all projects. The second part is a Graphical User Interface (GUI) which is used to create reports and to add, update, or delete projects. Both programs currently support Windows 95 and Windows NT.

Command Line Options

Generation of project documentation is done at the command line. ADMP has the following command line options:

-rA
If the –p option is given, ADMP will refresh all information for the given project. If this option is given alone, ADMP will refresh all information for all defined projects.

-rf filename
The –p option must be given with this option. This option will refresh all information for the given file.

-RF
The –p option must be given with this option. This option will refresh only the file list for the given project.

-g phrase
If the –p option is given, ADMP will search all files in the given project for the specified phrase. If this option is given alone, ADMP will search all files for the specified phrase.

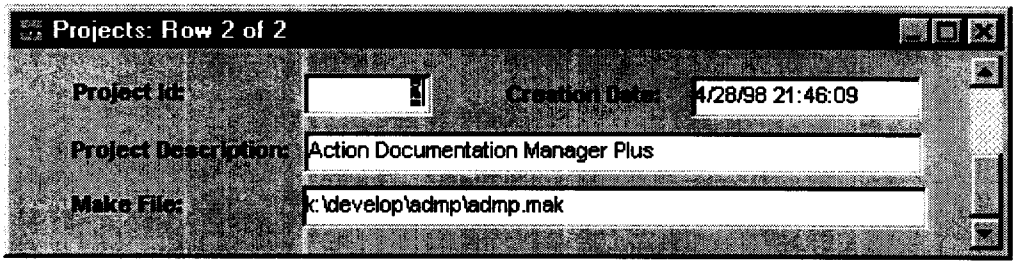
-p project
This option is used to specify the project.

For example, the following command would be used to refresh all documentation for project 2:

- admp –rA –p 2

Adding, Updating, and Deleting Projects

ADMP provides a GUI interface for adding updating, and deleting. The screen for adding, updating, and deleting projects follows:



Reports

Currently, ADMP includes the following reports:

- Function List – This report shows a listing of all functions defined in projects that are tracked by ADMP. For each function, this reports lists the project where the function can be found, the file and line in that file where the function can be found, the non-comment/non-blank line count of the function, and the comment line count of the function. This report can be queried or sorted by any of

the fields in the report. This report is useful for tracking down functions that are defined in multiple places, for tracking down functions that are defined in one place but included in many projects, for checking to see if a function exists, and for finding the location of a function.

- Table References – This report shows a listing of all references to database tables. For each reference, this report displays the table(s) being referenced, the type of reference (update, delete, insert, or select), the name of the function referencing the table(s), the location of the function, and the project in which the function is included. This report can be queried or sorted by any of the fields in the report. This report is useful for projecting the impact of database modifications on the project, for making sure all needed changes are made to a project when a database change is made, and for limiting the number of functions that modify a table.
- Function Sheets – This report shows detailed information about a function. For each function, this report displays the name of the function, the location of the function; the name, identifier, and path of the project, and the function header. This report is useful for checking to see if a function exists and for checking to see what a function was designed to do.

The following section contains samples of the reports included with ADMP. The reports in the following section were generated by ADMP on the ADMP project! It should be noted that these three reports represent only a subset of the reports that could be created. Reports can be easily customized to the needs of a particular project or a particular development team.

SECTION II – Sample Reports

Function List

Function Name	Project Id	File Path	Line Id	Ncnb Line Count	Comment Line Count
AddFileToProject	2	/develop/admp/admpsql.sqc	250	51	4
AddFunctionToProject	2	/develop/admp/admpsql2.sqc	209	32	0
FindFunctionDefinition	2	/develop/admp/admpsql2.sqc	21	67	7
FindTableRefs	2	/develop/admp/admpsql.sqc	779	60	3
GetCommentAndNumLines	2	/develop/admp/admpsql2.sqc	388	71	24
GetFunctionId	2	/develop/admp/admpsql2.sqc	123	29	0
GetNCNBlineCount	2	/develop/admp/admpsql.sqc	571	103	4
GetNextFileId	2	/develop/admp/admpsql.sqc	213	19	1
GetNextFunctionId	2	/develop/admp/admpsql2.sqc	169	20	1
InsertNewDeleteStatement	2	/develop/admp/admpsql.sqc	964	70	3
InsertNewInsertStatement	2	/develop/admp/admpsql.sqc	1146	64	3
InsertNewSelectStatement	2	/develop/admp/admpsql.sqc	868	70	3
InsertNewUpdateStatement	2	/develop/admp/admpsql.sqc	1062	58	3
main	2	/develop/admp/admp.c	20	146	1
RefreshAllFunctions	2	/develop/admp/admpsql2.sqc	717	56	2
RefreshFunctionHeader	2	/develop/admp/admpsql2.sqc	262	28	73
RefreshProjectFile	2	/develop/admp/admpsql2.sqc	503	155	30
RefreshProjectFileList	2	/develop/admp/admpsql.sqc	330	166	29
SkipFunctionBody	2	/develop/admp/admpsql.sqc	712	26	9

Table References

Table Name	Action Id	Function Name	File Path	Project Id	Line Id
CFG_FUNCTION_DEFS	INSERT	AddFunctionToProject	/develop/admp/admpsql2.sqc	2	209
CFG_FUNCTION_DEFS	SELECT	GetFunctionId	/develop/admp/admpsql2.sqc	2	123
CFG_FUNCTION_DEFS	SELECT	GetNextFunctionId	/develop/admp/admpsql2.sqc	2	169
CFG_FUNCTION_DEFS	DELETE	RefreshProjectFile	/develop/admp/admpsql2.sqc	2	503
CFG_FUNCTION_DEFS	DELETE	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_FUNCTION_DEFS	UPDATE	SkipFunctionBody	/develop/admp/admpsql.sqc	2	712
CFG_FUNCTION_HEADER	INSERT	RefreshFunctionHeader	/develop/admp/admpsql2.sqc	2	262
CFG_FUNCTION_HEADER	DELETE	RefreshProjectFile	/develop/admp/admpsql2.sqc	2	503
CFG_FUNCTION_HEADER	DELETE	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_FUNCTION_MODS	DELETE	RefreshProjectFile	/develop/admp/admpsql2.sqc	2	503
CFG_FUNCTION_MODS	DELETE	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_FUNCTION_REFS	DELETE	RefreshProjectFile	/develop/admp/admpsql2.sqc	2	503
CFG_FUNCTION_REFS	DELETE	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_PROJECT	SELECT	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_PROJECT	SELECT	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_PROJECT_FILES	INSERT	AddFileToProject	/develop/admp/admpsql.sqc	2	250
CFG_PROJECT_FILES	SELECT	GetNextFileId	/develop/admp/admpsql.sqc	2	213
CFG_PROJECT_FILES	SELECT	RefreshAllFunctions	/develop/admp/admpsql2.sqc	2	717
CFG_PROJECT_FILES	SELECT	RefreshAllFunctions	/develop/admp/admpsql2.sqc	2	717
CFG_PROJECT_FILES	SELECT	RefreshProjectFile	/develop/admp/admpsql2.sqc	2	503
CFG_PROJECT_FILES	UPDATE	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_PROJECT_FILES	SELECT	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_PROJECT_FILES	DELETE	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330
CFG_TABLE_REFS	INSERT	InsertNewDeleteStatement	/develop/admp/admpsql.sqc	2	964
CFG_TABLE_REFS	INSERT	InsertNewInsertStatement	/develop/admp/admpsql.sqc	2	1146
CFG_TABLE_REFS	INSERT	InsertNewSelectStatement	/develop/admp/admpsql.sqc	2	868
CFG_TABLE_REFS	INSERT	InsertNewUpdateStatement	/develop/admp/admpsql.sqc	2	1062
CFG_TABLE_REFS	DELETE	RefreshProjectFile	/develop/admp/admpsql2.sqc	2	503
CFG_TABLE_REFS	DELETE	RefreshProjectFileList	/develop/admp/admpsql.sqc	2	330

Function Name: **main** Ncnb Line Count: 146

File Path: /develop/admp/admp.c Line Id: 20

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************************
* PROGRAM: admp
* PURPOSE: the purpose of this program is to generate documentation
*    for a given project.
* ARGS:    the following command line args can be used
*   admp [-rA|-rf filename|-rF] [-g phrase] [-p pid]
*
*   -rA  -- refresh --
*     -- everything for a project if a pid is specified
*     -- everything for all projects if no pid is specified
*   -rf  filename -- full path of file to refresh for this pid
*   -RF  FILELIST -- refresh file list only for this pid
*   -g  -- search all files in the project for a phrase if a pid is given
*    -- search all files if no pid is given
*   -p  -- pid -- project identifier
*****************************************************************************/
```

Function Name: **FindFunctionDefinition** Ncnb Line Count: 67

File Path: /develop/admp/admpsql2.sqc Line Id: 21

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  FindFunctionDefinition
* PURPOSE:   Given a pointer to a file which is open to the
*     second line of a function definition, extract the
*     name of the function.
* PARAMETERS:  fp -- the file pointer
*     FirstLIne  -- the first line of text of the function def
*     function_name -- the name of the function (this will
*      be returned )
*     NumOfLines -- the number of lines we have moved forward
*      in the file....if we are successful this will be 0
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- best guess is this isn?t a function after all
* NOTES:  if we fail and don?t think its a function, the file pointer will
*    be advanced past whatever it is.  If we are successful the
*    file pointer will be left as it was to start with.
* AUTHOR: DWM
*****************************************************************/
```

Function Name: **GetFunctionId** Ncnb Line Count: 29

File Path: /develop/admp/admpsql2.sqc Line Id: 123

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/************************************************************
* FUNCTION:
ctionId
* PURPOSE:  lookup the FunctionId for this function within the
*   given project
* PARAMETERS:
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
*   NOTFOUND_CODE -- function not in the project
* NOTES:
* AUTHOR: DWM
************************************************************/
```

Function Name: **GetNextFunctionId** Ncnb Line Count: 20

File Path: /develop/admp/admpsql2.sqc Line Id: 169

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/************************************************************************
* FUNCTION:  GetNextFunctionId
* PURPOSE:   This function gets the next unique function identifier
*     available for the given project.
* PARAMETERS: project_id -- the project for which we need a new
*      file_id
*    function_id -- the file_id we find and pass back
* RETURN:  SUCCESS_CODE -- we got a new file_id
*     FAILURE_CODE -- an error occured
* AUTHOR:  DWM
* NOTE:
************************************************************************/
```

Function Name: **AddFunctionToProject** Ncnb Line Count: 32

File Path: /develop/admp/admpsql2.sqc Line Id: 209

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  AddFunctionToProject
* PURPOSE:  This function adds a function to the project
*    and returns its function_id
* PARAMETERS: project_id -- the project to add the function to
*   file_id  -- the file where the function is found
*   function_name -- the name of the function
*   line_id -- the line on which the function starts
*   function_id -- the unique function identifier assigned
*    to this function and passed back to the calling function
* RETURN:  SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR:  DWM
*****************************************************************/
```

Function Name: **RefreshFunctionHeader** Ncnb Line Count: 28

File Path: /develop/admp/admpsql2.sqc Line Id: 262

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  RefreshFunctionHeader
* PURPOSE:  This function puts the function header found in
*    comment into the database
* PARAMETERS: project_id -- the project we are dealing with
*   file_id  -- the file where the function is found
*   function_id -- the id of the function
*   comment -- the function header
*   CommentLen -- some unused argument
* RETURN:  SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR:  DWM
*****************************************************************/
```

Function Name: **GetCommentAndNumLines** Ncnb Line Count: 71

File Path: /develop/admp/admpsql2.sqc Line Id: 388

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  GetCommentAndNumLines
* PURPOSE:  This function grabs a comment block from the file
*    the file must open and the file pointer sitting
*    at the second line of the comment block.  The
*    first line of the comment block is passed in.
*    The comment length in number of lines is passed
*    back
* PARAMETERS: fp -- the file pointer
*   FirstLine -- the firstline of the comment block
*   comment -- the comment block we found
*   CommentLength -- the length of the block we found in lines
* RETURN:  SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR:  DWM
*****************************************************************/
```

Function Name: **RefreshProjectFile** Ncnb Line Count: 155

File Path: /develop/admp/admpsql2.sqc Line Id: 503

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  RefreshProjectFile
* PURPOSE:  Given a project and a file, refresh the function headers
*  function definitions, function references, table references,
*  and modification history for the functions found in that file
* PARAMETERS: project_id -- the project whose file list we should
*   refresh
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*****************************************************************/
```

Function Name: **RefreshAllFunctions** Ncnb Line Count: 56

File Path: /develop/admp/admpsql2.sqc Line Id: 717

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  RefreshAllFunctions
* PURPOSE:  Given a project, refresh the function headers
*  function definitions, function references, table references,
*  and modification history for all functions found in the project
* PARAMETERS: project_id -- the project whose functions we should
*     refresh
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*****************************************************************/
```

Function Name: **GetNextFileId** Ncnb Line Count: 19

File Path: /develop/admp/admpsql.sqc Line Id: 213

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/************************************************************************
All these will be elsewhere later
************************************************************************/
/*
#define SUCCESS_CODE 1
#define FAILURE_CODE 2
#define NOTFOUND_CODE 3


#define FALSE 0
#define TRUE 1
*/
/*****************************************************************************
* FUNCTION: Connect
* PURPOSE: connect to the database
* RETURNS: none
* PARAMTERS: none
* AUTHOR:  DWM
*****************************************************************************/
/*void Connect()
{
EXEC SQL begin declare section;
char database[40]=&danieljr.admp&;
EXEC SQL end declare section;
EXEC SQL CONNECT TO :database USER sa;
```

Function Name: **AddFileToProject** Ncnb Line Count: 51

File Path: /develop/admp/admpsql.sqc Line Id: 250

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  AddFileToProject
* PURPOSE:   This function adds a file with path file_path to
*     the project identified by project_id.  It does
*     this by inserting into the cfg_project_files
*     table.
* PARAMETERS: project_id -- the project we are adding a file to
*    file_path -- the full path of the file
* RETURN:  SUCCESS_CODE -- we got a new file_id
*     FAILURE_CODE -- an error occured
* AUTHOR:  DWM
* NOTE:
*****************************************************************/
```

Function Name: **RefreshProjectFileList** Ncnb Line Count: 166

File Path: /develop/admp/admpsql.sqc Line Id: 330

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  RefreshProjectFileList
* PURPOSE:  Given a particular project, refresh the file list
*  found in the cfg_project_files list.  As we refresh, we
*  delete files from our list that are no longer there, add
*  files that are new, and update the last refresh time on
*  files we already new about.
* PARAMETERS: project_id -- the project whose file list we should
*   refresh
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*****************************************************************/
```

Function Name: **GetNCNBlineCount** Ncnb Line Count: 103

File Path: /develop/admp/admpsql.sqc Line Id: 571

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************************
* FUNCTION:  GetNCNBlineCount
* PURPOSE:  Get the number of non-comment, non-blank lines in the function
*  which starts on the line before current position
*  in the open file refered to by fp
* PARAMETERS: fp -- pointer to open file where function is
*    FirstLine -- the line of text above the current position in the
*     file
*    NCNBlineCount -- the count of NCNB lines that is returned
*    CommentLineCount -- the count of comment lines in the function that
*     is returned to the calling function
*    FunLen  -- the total number of lines in the function that is returned
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*****************************************************************************/
```

Function Name: **SkipFunctionBody** Ncnb Line Count: 26

File Path: /develop/admp/admpsql.sqc Line Id: 712

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  SkipFunctionBody
* PURPOSE:  Given a file pointer to an open file at the second line
*    of a fuction definition fill in the non-comment, non-blank
*    line counts for the function and advance the file pointer
*    past the function
* PARAMETERS: fp -- the file pointer
*    FirstLine -- the first of the function definition
*    project_id -- the project this file belongs to
*    file_id -- the identifier of the file
*    function_id -- the ID of this function which is unique within
*     this project
*    FunctionLen -- the length of the function -1 in lines
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*****************************************************************/
```

Function Name: **FindTableRefs** Ncnb Line Count: 60

File Path: /develop/admp/admpsql.sqc Line Id: 779

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:  FindTableRefs
* PURPOSE:  Check to see if the line we are on has an sql statement on it
*    if it does, fill in the cfg_table_refs table with the
*    project, file, function, table_name and whether it is
*    a select, an update, an insert or a delete
* PARAMETERS: fp -- the file pointer
*   buffer -- the line of text we are looking at in the file
*   project_id -- the project this file belongs to
*   file_id -- the identifier of the file
*   function_id -- the ID of this function which is unique within
*    this project
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
*   NOTFOUND_CODE -- this is not a sql statement
* NOTES:
* AUTHOR: DWM
*****************************************************************/
```

Function Name: **InsertNewSelectStatement** Ncnb Line Count: 70

File Path: /develop/admp/admpsql.sqc Line Id: 868

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*****************************************************************
* FUNCTION:   InsertNewSelectStatement
* PURPOSE:   Check to see if the SQLstatement is a select statement
*     if it iss, fill in the cfg_table_refs table with the
*     project, file, function, table_name and ?SELECT?
* PARAMETERS: project_id -- the project this file belongs to
*    file_id -- the identifier of the file
*    function_id -- the ID of this function which is unique within
*     this project
*    SQLstatement -- the sql statement we are dealing with
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*****************************************************************/
```

Function Name: **InsertNewDeleteStatement** Ncnb Line Count: 70

File Path: /develop/admp/admpsql.sqc Line Id: 964

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*************************************************************
* FUNCTION:   InsertNewDeleteStatement
* PURPOSE:   Check to see if the SQLstatement is a delete statement
*     if it iss, fill in the cfg_table_refs table with the
*     project, file, function, table_name and ?DELETE?
* PARAMETERS: project_id -- the project this file belongs to
*    file_id -- the identifier of the file
*    function_id -- the ID of this function which is unique within
*     this project
*    SQLstatement -- the sql statement we are dealing with
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*************************************************************/
```

Function Name: **InsertNewUpdateStatement** Ncnb Line Count: 58

File Path: /develop/admp/admpsql.sqc Line Id: 1062

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

HEADER

```
/*************************************************************
* FUNCTION:   InsertNewUpdateStatement
* PURPOSE:   Check to see if the SQLstatement is a update statement
*     if it iss, fill in the cfg_table_refs table with the
*     project, file, function, table_name and ?UPDATE?
* PARAMETERS: project_id -- the project this file belongs to
*    file_id -- the identifier of the file
*    function_id -- the ID of this function which is unique within
*     this project
*    SQLstatement -- the sql statement we are dealing with
* RETURN: SUCCESS_CODE -- it worked
*   FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*************************************************************/
```

Function Name: **InsertNewInsertStatement** Ncnb Line Count: 64

File Path: /develop/admp/admpsql.sqc Line Id: 1146

Project Id: Action Documentation Manager Plus 2

k:\develop\admp\admp.mak

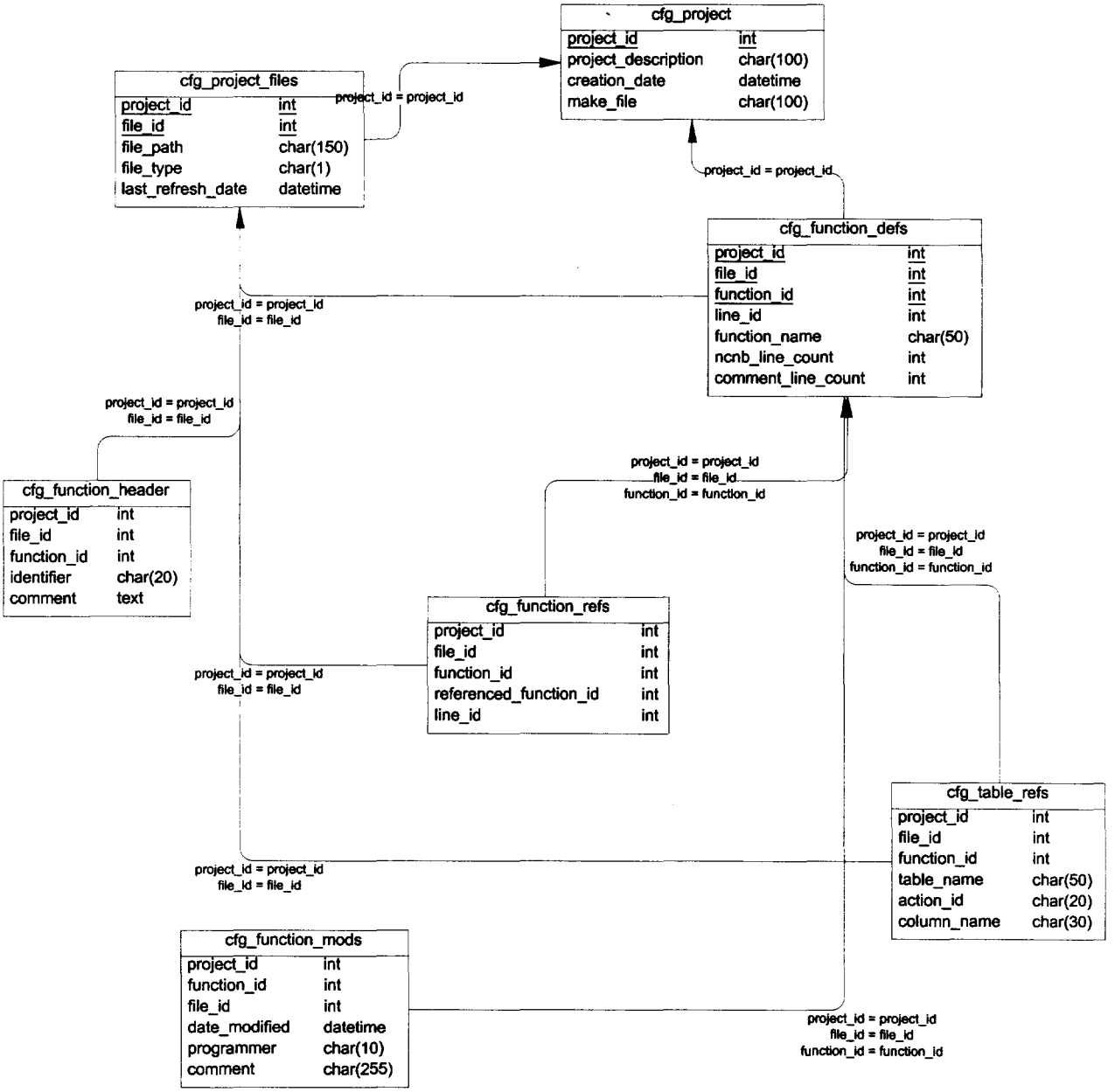
HEADER

```
/*************************************************************
* FUNCTION:  InsertNewInsertStatement
* PURPOSE:  Check to see if the SQLstatement is a insert statement
*    if it iss, fill in the cfg_table_refs table with the
*    project, file, function, table_name and ?INSERT?
* PARAMETERS: project_id -- the project this file belongs to
*   file_id -- the identifier of the file
*   function_id -- the ID of this function which is unique within
*    this project
*   SQLstatement -- the sql statement we are dealing with
* RETURN: SUCCESS_CODE -- it worked
*  FAILURE_CODE -- there was an error
* NOTES:
* AUTHOR: DWM
*************************************************************/
```

SECTION III – Database Design

PDM Graphs

Global Model Graph



Lists of Objects

User List

Name	Code
dbo	dbo

Table List

Name	Code	Number
cfg_function_defs	cfg_function_defs	
cfg_function_header	cfg_function_header	
cfg_function_mods	cfg_function_mods	
cfg_function_refs	cfg_function_refs	
cfg_project	cfg_project	
cfg_project_files	cfg_project_files	
cfg_table_refs	cfg_table_refs	

Reference List

Reference Name	Reference Code	Parent	Child
Ref_13	REF_13	cfg_project	cfg_project_files
Ref_24	REF_24	cfg_project_files	cfg_function_defs
Ref_29	REF_29	cfg_project	cfg_function_defs
Ref_38	REF_38	cfg_project_files	cfg_function_header
Ref_50	REF_50	cfg_function_defs	cfg_function_mods
Ref_63	REF_63	cfg_function_defs	cfg_function_refs
Ref_70	REF_70	cfg_project_files	cfg_function_refs
Ref_82	REF_82	cfg_function_defs	cfg_table_refs
Ref_89	REF_89	cfg_project_files	cfg_table_refs

Column List

Column Code	Type
action_id	char(20)
column_name	char(30)
comment	text
comment	char(255)
comment_line_count	int
creation_date	datetime
date_modified	datetime
file_id	int
file_id	int
file_id	int
file_id	int
file_id	int
file_id	int
file_path	char(150)
file_type	char(1)
function_id	int
function_id	int
function_id	int

Column Code	Type
function_id	int
function_id	int
function_name	char(50)
identifier	char(20)
last_refresh_date	datetime
line_id	int
line_id	int
make_file	char(100)
ncnb_line_count	int
programmer	char(10)
project_description	char(100)
project_id	int
project_id	int
project_id	int
project_id	int
project_id	int
project_id	int
project_id	int
referenced_function_id	int
table_name	char(50)

Tables Information

Table cfg_function_defs

Name:	cfg_function_defs
Code:	cfg_function_defs
Label:	
Owner:	dbo
Number:	
PK constraint:	
Source:	

Column List

Name	Code	Type	P	M
project_id	project_id	int	Yes	Yes
file_id	file_id	int	Yes	Yes
function_id	function_id	int	Yes	Yes
line_id	line_id	int	No	Yes
function_name	function_name	char(50)	No	Yes
ncnb_line_count	ncnb_line_count	int	No	No
comment_line_count	comment_line_count	int	No	No

Fill List

Name	Type	Null	Distinct	Length	M	U	F
project_id	int	0%	100%	0	Yes	No	Yes
file_id	int	0%	100%	0	Yes	No	Yes
function_id	int	0%	100%	0	Yes	No	No
line_id	int	0%	100%	0	Yes	No	No
function_name	char(50)	0%	100%	50	Yes	No	No
ncnb_line_count	int	0%	100%	0	No	No	No
comment_line_count	int	0%	100%	0	No	No	No

Reference to List

Primary Key	Reference to	Foreign Key
project_id	cfg_project_files	project_id
file_id		file_id
project_id	cfg_project	project_id

Reference by List

Referenced by	Primary Key	Foreign Key
cfg_function_mods	project_id	project_id
	file_id	file_id
	function_id	function_id
cfg_function_refs	project_id	project_id
	file_id	file_id
	function_id	function_id
cfg_table_refs	project_id	project_id

Referenced by	Primary Key	Foreign Key
	file_id function_id	file_id function_id

Table cfg_function_header

Name: cfg_function_header
Code: cfg_function_header
Label:
Owner: dbo
Number:
PK constraint:
Source:

Column List

Name	Code	Type	P	M
project_id	project_id	int	No	Yes
file_id	file_id	int	No	Yes
function_id	function_id	int	No	Yes
identifier	identifier	char(20)	No	Yes
comment	comment	text	No	Yes

Fill List

Name	Type	Null	Distinct	Length	M	U	F
project_id	int	0%	100%	0	Yes	No	Yes
file_id	int	0%	100%	0	Yes	No	Yes
function_id	int	0%	100%	0	Yes	No	No
identifier	char(20)	0%	100%	20	Yes	No	No
comment	text	0%	100%	0	Yes	No	No

Reference to List

Primary Key	Reference to	Foreign Key
project_id file_id	cfg_project_files	project_id file_id

Table cfg_function_mods

Name: cfg_function_mods
Code: cfg_function_mods
Label:
Owner: dbo
Number:
PK constraint:
Source:

Column List

Name	Code	Type	P	M
project_id	project_id	int	No	Yes
function_id	function_id	int	No	Yes
file_id	file_id	int	No	Yes
date_modified	date_modified	datetime	No	Yes
programmer	programmer	char(10)	No	Yes
comment	comment	char(255)	No	Yes

Fill List

Name	Type	Null	Distinct	Length	M	U	F
project_id	int	0%	100%	0	Yes	No	Yes
function_id	int	0%	100%	0	Yes	No	Yes
file_id	int	0%	100%	0	Yes	No	Yes
date_modified	datetime	0%	100%	0	Yes	No	No
programmer	char(10)	0%	100%	10	Yes	No	No
comment	char(255)	0%	100%	255	Yes	No	No

Reference to List

Primary Key	Reference to	Foreign Key
project_id file_id function_id	cfg_function_defs	project_id file_id function_id

Table cfg_function_refs

Name: cfg_function_refs
Code: cfg_function_refs
Label:
Owner: dbo
Number:
PK constraint:
Source:

Column List

Name	Code	Type	P	M
project_id	project_id	int	No	Yes
file_id	file_id	int	No	Yes
function_id	function_id	int	No	Yes
referenced_function_id	referenced_function_id	int	No	Yes
line_id	line_id	int	No	No

Fill List

Name	Type	Null	Distinct	Length	M	U	F
project_id	int	0%	100%	0	Yes	No	Yes
file_id	int	0%	100%	0	Yes	No	Yes
function_id	int	0%	100%	0	Yes	No	Yes

Name	Type	Null	Distinct	Length	M	U	F
referenced_function_id	int	0%	100%	0	Yes	No	No
line_id	int	0%	100%	0	No	No	No

Reference to List

Primary Key	Reference to	Foreign Key
project_id	cfg_function_defs	project_id
file_id		file_id
function_id		function_id
project_id	cfg_project_files	project_id
file_id		file_id

Table cfg_project

Name: cfg_project
Code: cfg_project
Label:
Owner: dbo
Number:
PK constraint:
Source:

Column List

Name	Code	Type	P	M
project_id	project_id	int	Yes	Yes
project_description	project_description	char(100)	No	No
creation_date	creation_date	datetime	No	Yes
make_file	make_file	char(100)	No	Yes

Fill List

Name	Type	Null	Distinct	Length	M	U	F
project_id	int	0%	100%	0	Yes	Yes	No
project_description	char(100)	0%	100%	100	No	No	No
creation_date	datetime	0%	100%	0	Yes	No	No
make_file	char(100)	0%	100%	100	Yes	No	No

Reference by List

Referenced by	Primary Key	Foreign Key
cfg_project_files	project_id	project_id
cfg_function_defs	project_id	project_id

Table cfg_project_files

Name: cfg_project_files
Code: cfg_project_files

Label:
Owner: dbo
Number:
PK constraint:
Source:

Column List

Name	Code	Type	P	M
project_id	project_id	int	Yes	Yes
file_id	file_id	int	Yes	Yes
file_path	file_path	char(150)	No	Yes
file_type	file_type	char(1)	No	Yes
last_refresh_date	last_refresh_date	datetime	No	No

Fill List

Name	Type	Null	Distinct	Length	M	U	F
project_id	int	0%	100%	0	Yes	No	Yes
file_id	int	0%	100%	0	Yes	No	No
file_path	char(150)	0%	100%	150	Yes	No	No
file_type	char(1)	0%	100%	1	Yes	No	No
last_refresh_date	datetime	0%	100%	0	No	No	No

Reference to List

Primary Key	Reference to	Foreign Key
project_id	cfg_project	project_id

Reference by List

Referenced by	Primary Key	Foreign Key
cfg_function_defs	project_id file_id	project_id file_id
cfg_function_header	project_id file_id	project_id file_id
cfg_function_refs	project_id file_id	project_id file_id
cfg_table_refs	project_id file_id	project_id file_id

Table cfg_table_refs

Name: cfg_table_refs
Code: cfg_table_refs
Label:
Owner: dbo
Number:
PK constraint:
Source:

Column List

Name	Code	Type	P	M
project_id	project_id	int	No	Yes
file_id	file_id	int	No	Yes
function_id	function_id	int	No	Yes
table_name	table_name	char(50)	No	Yes
action_id	action_id	char(20)	No	Yes
column_name	column_name	char(30)	No	No

Fill List

Name	Type	Null	Distinct	Length	M	U	F
project_id	int	0%	100%	0	Yes	No	Yes
file_id	int	0%	100%	0	Yes	No	Yes
function_id	int	0%	100%	0	Yes	No	Yes
table_name	char(50)	0%	100%	50	Yes	No	No
action_id	char(20)	0%	100%	20	Yes	No	No
column_name	char(30)	0%	100%	30	No	No	No

Reference to List

Primary Key	Reference to	Foreign Key
project_id	cfg_function_defs	project_id
file_id		file_id
function_id		function_id
project_id	cfg_project_files	project_id
file_id		file_id

```
CREATE TABLE dbo.cfg_project (
project_id int NOT NULL ,
project_description char (100) NULL ,
creation_date datetime NOT NULL CONSTRAINT DF__cfg_proje__creat__0B71BE83 DEFAU
make_file char (100) NOT NULL ,
CONSTRAINT PK_cfg_project_1__10 PRIMARY KEY  CLUSTERED
(
 project_id
),
CONSTRAINT UQ_cfg_project_2__10 UNIQUE  NONCLUSTERED
(
 make_file
)
)
GO

CREATE TABLE dbo.cfg_project_files (
project_id int NOT NULL ,
file_id int NOT NULL ,
file_path char (150) NOT NULL ,
file_type char (1) NOT NULL ,
last_refresh_date datetime NULL CONSTRAINT DF_cfg_projec_last_refres4__11 DEFAU
CONSTRAINT PK_cfg_project_files_1__11 PRIMARY KEY  CLUSTERED
(
 project_id,
 file_id
),
CONSTRAINT UQ_cfg_project_files_2__11 UNIQUE  NONCLUSTERED
(
 project_id,
 file_path
),
CONSTRAINT FK_cfg_project_files_3__11 FOREIGN KEY
(
 project_id
) REFERENCES dbo.cfg_project (
 project_id
)
)
GO

CREATE TABLE dbo.cfg_function_defs (
project_id int NOT NULL ,
file_id int NOT NULL ,
function_id int NOT NULL ,
line_id int NOT NULL ,
function_name char (50) NOT NULL ,
ncnb_line_count int NULL ,
comment_line_count int NULL ,
CONSTRAINT PK_cfg_function_defs_1__10 PRIMARY KEY  CLUSTERED
(
 project_id,
 file_id,
 function_id
),
CONSTRAINT UQ_cfg_function_defs_2__10 UNIQUE  NONCLUSTERED
(
 project_id,
```

```
 file_id,
 function_name
),
CONSTRAINT FK_cfg_function_defs_1__11 FOREIGN KEY
(
 project_id,
 file_id
) REFERENCES dbo.cfg_project_files (
 project_id,
 file_id
),
CONSTRAINT FK_cfg_function_defs_3__10 FOREIGN KEY
(
 project_id
) REFERENCES dbo.cfg_project (
 project_id
)
)
GO

CREATE TABLE dbo.cfg_function_header (
project_id int NOT NULL ,
file_id int NOT NULL ,
function_id int NOT NULL ,
identifier char (20) NOT NULL ,
comment text NOT NULL ,
CONSTRAINT PK_cfg_function_header_1__10 PRIMARY KEY  CLUSTERED
(
 project_id,
 file_id,
 function_id,
 identifier
),
CONSTRAINT FK_cfg_function_header_1__10 FOREIGN KEY
(
 project_id,
 file_id,
 function_id
) REFERENCES dbo.cfg_function_defs (
 project_id,
 file_id,
 function_id
),
CONSTRAINT FK_cfg_function_header_1__11 FOREIGN KEY
(
 project_id,
 file_id
) REFERENCES dbo.cfg_project_files (
 project_id,
 file_id
)
)
GO

CREATE TABLE dbo.cfg_function_mods (
project_id int NOT NULL ,
function_id int NOT NULL ,
file_id int NOT NULL ,
date_modified datetime NOT NULL ,
programmer char (10) NOT NULL ,
```

```
comment char (255) NOT NULL ,
CONSTRAINT PK_cfg_function_mods_1__10 PRIMARY KEY  CLUSTERED
(
 file_id,
 function_id,
 programmer,
 date_modified,
 comment
),
CONSTRAINT FK_cfg_function_mods_1__11 FOREIGN KEY
(
 project_id,
 file_id
) REFERENCES dbo.cfg_project_files (
 project_id,
 file_id
),
CONSTRAINT FK_cfg_function_mods_2__10 FOREIGN KEY
(
 project_id,
 file_id,
 function_id
) REFERENCES dbo.cfg_function_defs (
 project_id,
 file_id,
 function_id
)
)
GO

CREATE TABLE dbo.cfg_function_refs (
project_id int NOT NULL ,
file_id int NOT NULL ,
function_id int NOT NULL ,
referenced_function_id int NOT NULL ,
line_id int NULL ,
CONSTRAINT FK_cfg_function_refs_1__10 FOREIGN KEY
(
 project_id,
 file_id,
 function_id
) REFERENCES dbo.cfg_function_defs (
 project_id,
 file_id,
 function_id
),
CONSTRAINT FK_cfg_function_refs_1__11 FOREIGN KEY
(
 project_id,
 file_id
) REFERENCES dbo.cfg_project_files (
 project_id,
 file_id
)
)
GO

CREATE TABLE dbo.cfg_table_refs (
project_id int NOT NULL ,
file_id int NOT NULL ,
```

```
function_id int NOT NULL ,
table_name char (50) NOT NULL ,
action_id char (20) NOT NULL ,
column_name char (30) NULL ,
CONSTRAINT FK_cfg_table_refs_1__10 FOREIGN KEY
(
 project_id,
 file_id,
 function_id
) REFERENCES dbo.cfg_function_defs (
 project_id,
 file_id,
 function_id
),
CONSTRAINT FK_cfg_table_refs_1__11 FOREIGN KEY
(
 project_id,
 file_id
) REFERENCES dbo.cfg_project_files (
 project_id,
 file_id
)
)
GO
```